

Instruções, dados e expressões

Marina Andretta

ICMC-USP

1 de março de 2016

Um computador manipula informações presentes em sua memória. Estas informações estão divididas em dois tipos:

- 1 **Instruções**: comandos que determinam o funcionamento do computador e como os dados devem ser tratados.
- 2 **Dados**: informações que devem ser manipuladas pelos computador.

Tipos de dados

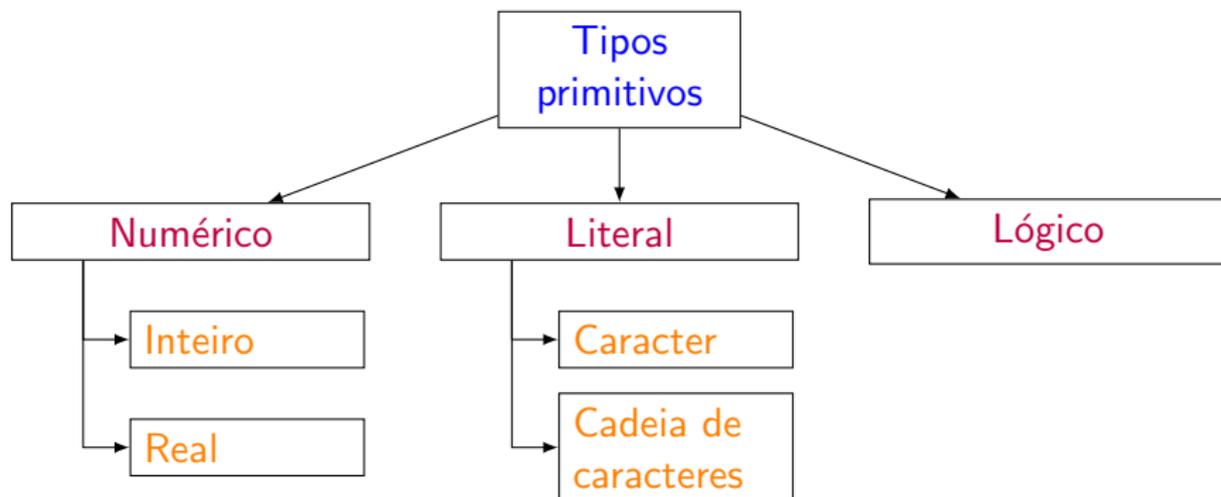
Podemos dizer que o objetivo de um computador é resolver tarefas. Para isso, ele manipula dados.

Por exemplo, uma nota, um nome, um endereço são dados que podem ser manipulados por um computador.

Os dados podem ser classificados em **tipos**. Tipos de dados diferentes são armazenados e manipulados pelo computador de maneiras diferentes.

Definir o tipo de um dado é útil para, além de classificar o dado, definir a quantidade de memória necessária para armazená-lo.

Os tipos de dados podem ser classificados da seguinte forma:



Tipos de dados: inteiro

Um dado numérico pertencente ao conjunto dos números inteiros é classificado como **inteiro**.

Alguns exemplos de dados inteiros são:

- Já li **1000** livros.
- O resultado da conta é **-7**.
- **42**.

Na linguagem C, os dados inteiros podem ser definidos usando **short int**, **int**, **long int**, **long long int**, **unsigned short int**, **unsigned int**, **unsigned long int** ou **unsigned long long int**.

Tipos de dados: real

Um dado numérico que pertence ao conjunto dos números reais é classificado como **real**.

Alguns exemplos de dados reais são:

- Minha média na disciplina foi **6,9**.
- A distância para a minha casa é de **7,13** quilômetros.
- O saldo da minha conta no banco é de **-42,42** reais.

Na linguagem C, os dados reais podem ser definidos usando **float**, **double** ou **long double**.

Tipos de dados: real

Dados reais, especialmente quando muito grandes ou muito pequenos, podem ser descritos usando o que chamamos de notação de ponto flutuante (similar a notação científica).

Nesta notação, o número correspondente a $\alpha \times 10^\beta$ é representado por $\alpha E\beta$.

Por exemplo, o número $1,564 \times 10^{-17}$ é representado por $1.564E-17$.

Tipos de dados: caracter

Um dado que contém apenas um símbolo, que pode ser uma letra, um dígito ou um símbolo especial, é classificado como **caracter**.

Alguns exemplos de caracteres são:

- Meu conceito na disciplina foi **B**.
- O número sete é representado pelo símbolo **7**.
- As perguntas terminam com **?**.

Na linguagem C, os caracteres podem ser definidos usando **char** ou **unsigned char**.

Tipos de dados: cadeia de caracteres

Se um dado contém vários símbolos, que podem ser letras, dígitos ou um símbolos especiais, é classificado como **cadeia de caracteres** ou *string*.

Alguns exemplos de cadeias de caracteres são:

- Ele sempre me fala “**eu te disse**”.
- Quando a vida te coloca pra baixo, **continue a nadar**.
- Esta prova está **?!#@**.

Na linguagem C, as cadeias de caracteres podem ser definidos usando **char[]** ou **unsigned char[]**.

Tipos de dados: lógico

Se um dado contém apenas dois possíveis valores, pode ser classificado como **lógico**.

Alguns exemplos de dados lógicos são:

- O telefone está **quebrado** ou **funcionando**.
- A sua frase é **verdadeira** ou **falsa**.
- A sacola está **aberta** ou **fechada**.

Na linguagem C não é possível definir dados do tipo lógico.

Tamanhos dos tipos de dados em linguagem C

Computadores e compiladores atuais, usando padrão ANSI C, possuem os seguintes valores típicos para os tamanhos dos tipos de dados.

Tipo	Tamanho (bytes)	Intervalo
char	1	-128 a 127
unsigned char	1	0 a 255
short int	2	-32.768 a 32.767
unsigned short int	2	0 a 65.535
int	4	-2.147.483.648 a 2.147.483.647
unsigned int	4	0 a 4.294.967.295
float	4	24 bits significativos
double	8	53 bits significativos

Variáveis

Para poder manipular dados, o computador precisa armazená-los. Isso é feito usando **variáveis**.

Uma variável contém, basicamente, quatro informações:

- 1 **Nome**: nome usado durante o algoritmo quando se faz referência à variável. O nome de cada variável é único no algoritmo.
- 2 **Tipo**: tipo do dado que a variável armazena.
- 3 **Conteúdo** ou **valor**: dado que é armazenado na variável, que deve ser compatível com seu tipo.
- 4 **Endereço**: posição da memória em que a variável está armazenada.

Para definir o nome e o tipo de uma variável, fazemos a chamada **declaração de variável**.

Em pseudo-código, faremos isso da seguinte maneira:

declaracoes

inteiro idade

real nota

caracter sexo

cadeia endereco

logico matriculado

Em linguagem C, as mesmas variáveis são declaradas da seguinte maneira:

inteiro idade	int idade;
real nota	float nota;
caracter sexo	char sexo;
cadeia endereco	char endereco[50];
logico matriculado	int matriculado;

Os nomes das variáveis podem conter letras (maiúsculas ou minúsculas), dígitos ou o caracter sublinhado (`_`). O primeiro caracter do nome não pode ser um dígito. Letras maiúsculas e minúsculas são diferenciadas entre si ($A \neq a$).

Não são permitidos nomes de variáveis repetidos ou palavras reservadas (correspondentes a comandos já existentes na sua linguagem).

Recomenda-se que o nome de uma variável seja significativo, ou seja, que tenha relação com a informação que ela irá armazenar.

Depois de declaradas as variáveis, elas não têm nenhum conteúdo definido, apenas seu nome, tipo e endereço de memória (que NÃO é definido pelo programador).

As instruções básicas de uma linguagem de programação são divididas em três:

- 1 comando de atribuição;
- 2 comando de entrada;
- 3 comando de saída.

Comando de atribuição

O comando de atribuição permite atribuir um valor a uma variável.

O valor deve ser compatível com o tipo da variável.

Em pseudo-código, usamos o símbolo \leftarrow .

Em linguagem C, usamos o símbolo $=$.

Comando de atribuição

Exemplos de atribuição de valores a variáveis, em pseudo-código, são:

```
idade ← 10
```

```
nota ← 5.2
```

```
sexo ← 'f'
```

```
endereco ← ‘‘P Sherman, 42, Wallaby Way, Sydney’’
```

```
matriculado ← verdadeiro
```

Em linguagem C, as mesmas atribuições são feitas da seguinte maneira:

```
idade = 10;  
nota = 5.2;  
sexo = 'f';  
strcpy(endereco, "P Sherman, 42, Wallaby Way, Sydney");  
matriculado = 1;
```

Comando de atribuição

É muito importante saber que quando um valor novo é atribuído a uma variável, seu valor antigo é apagado e substituído pelo novo.

Por exemplo, se tivermos os comandos

```
idade ← 10
```

```
idade ← 12
```

após o primeiro comando, a variável `idade` passa a ter o valor 10. Quando executamos o segundo comando, o valor da variável `idade` é substituído por 12.

Comando de entrada

O comando de entrada permite que o algoritmo obtenha uma informação do usuário, através de um dispositivo de entrada, e guarde esta informação em uma variável.

Novamente, o valor armazenado na variável deve ser compatível com seu tipo.

Usaremos a entrada padrão como sendo o teclado.

Em pseudo-código, a leitura de dados é feita usando o comando **leia**. Em linguagem C, o comando é o **scanf**.

Exemplos de leitura de dados, em pseudo-código, são:

leia(idade)

leia(nota)

leia(idade, nota)

leia(sexo)

leia(endereco)

leia(matriculado)

Em linguagem C, as mesmas leituras de dados são feitas da seguinte maneira:

```
scanf(“%d”, &idade);  
scanf(“%f”, &nota);  
scanf(“%d %f”, &idade, &nota);  
scanf(“%c”, &sexo);  
scanf(“%s”, endereco);  
scanf(“%d”, &matriculado);
```

Comando de entrada

Quando há um comando de entrada no algoritmo, sua execução é paralisada e espera-se que o usuário digite o dado.

Quando o usuário digita um ENTER, a execução recomeça e o que foi digitado antes do ENTER é armazenado na variável indicada.

Comando de saída

O comando de saída é usado para que o algoritmo forneça informações ao usuário, através de um dispositivo de saída. Nosso dispositivo padrão será a tela.

Podem ser impressos na tela mensagens, valores de variáveis e resultados de expressões.

O comando de saída, em pseudo-código, é o **escreva**. Em linguagem C, o comando é **printf**.

Exemplos de impressão de dados, em pseudo-código, são:

```
escreva(idade)
```

```
escreva(‘‘Este programa calcula sua media\n’’)
```

```
escreva(‘‘A nota foi ’’, nota)
```

```
escreva(sexo, endereco)
```

```
escreva(matriculado)
```

Em linguagem C, as mesmas impressões de dados são feitas da seguinte maneira:

```
printf('‘%d’’, idade);  
printf('‘Este programa calcula sua media\n’’);  
printf('‘A nota foi %f’’, nota);  
printf('‘%c %s’’, sexo, endereco);  
printf('‘%d’’, matriculado);
```

Uma expressão numérica envolve constantes e o valor de variáveis numéricas.

Dois exemplos de expressão numérica são

$$\frac{4 - (3^5 - 1)}{13 + 2} \quad \text{e} \quad -\frac{x}{2} + (y^2 - 1).$$

Operadores numéricos

Os operadores numéricos são os seguintes:

Pseudo-código	Linguagem C	Operação
$a + b$	$a + b$	adição
$a - b$	$a - b$	subtração
$a * b$	$a * b$	multiplicação
a / b	a / b	divisão
$a :: b$		raiz a -ésima
$a ^ b$	<code>pow(a, b)</code>	potenciação
$a \text{ div } b$	a / b	divisão inteira
$a \text{ mod } b$	$a \% b$	resto da divisão inteira

Operadores numéricos

Os operadores `+`, `-`, `*` e `pow()` podem ser usados com operandos inteiros ou reais. Quando ao menos um dos operandos é real, o resultado também é real.

O operador `/` pode ser usado com operandos inteiros ou reais. Se ambos os operandos forem inteiros, na linguagem C, o resultado é o quociente da divisão do primeiro termo pelo segundo.

Os operadores `::`, `^`, `div (/)` e `mod (%)` devem ser usados com operandos inteiros.

Operadores numéricos

Se necessário, podem ser usados parênteses nas expressões numéricas.

As prioridades para as operações seguem as prioridades usuais da matemática.

Alguns exemplos de expressões numéricas em pseudo-código são:

$$(3^5 - 1)/(13 + 2) \quad \text{e} \quad -x/2 + y^2 - 1.$$

Em linguagem C, estas expressões são:

$$(\mathbf{pow}(3,5) - 1)/(13 + 2) \quad \text{e} \quad -x/2 + \mathbf{pow}(y,2) - 1.$$

Expressões lógicas são as que têm como resultado um valor lógico (verdadeiro ou falso).

Seus operandos podem ser constantes ou variáveis, lógicas ou numéricas.

Seus operadores podem ser lógicos, relacionais ou mesmo numéricos.

Parênteses podem ser usados nestas expressões. Na resolução destas expressões, é preciso primeiro resolver as expressões entre parênteses, seguidas das expressões numéricas.

Operadores relacionais

Operadores relacionais são utilizados para comparar valores numéricos e resultam em um valor lógico.

Os operadores relacionais são os seguintes:

Pseudo-código	Linguagem C	Operação
$a < b$	$a < b$	menor
$a \leq b$	$a \leq b$	menor ou igual
$a > b$	$a > b$	maior
$a \geq b$	$a \geq b$	maior ou igual
$a = b$	$a == b$	igual
$a \neq b$	$a != b$	diferente

Um exemplo de expressão lógica usando operadores relacionais é

$$(4 - 7) \geq (4 / 2)$$

Resolvendo esta expressão, temos

$$\begin{aligned}(4 - 7) &\geq (4 / 2) \\ -3 &\geq 2 \\ \mathbf{falso}\end{aligned}$$

Conectivos lógicos

Conectivos lógicos têm como operandos expressões lógicas e, como resultado, um valor lógico.

Os conectivos lógicos são os seguintes:

Pseudo-código	Linguagem C	Conectivo
a e b	a && b	e
a ou b	a b	ou
nao a	!a	não

A ordem de prioridade dos conectivos lógicos é **não**, **e**, **ou**.

Tabela verdade: não

A tabela verdade é uma tabela que associa todos os possíveis valores lógicos dos operandos em expressões com conectivos lógicos aos valores resultantes da expressão que usa este conectivo.

Para o conectivo lógico **não**, temos a seguinte tabela verdade:

a	nao a
verdadeiro	
falso	

Tabela verdade: não

A tabela verdade é uma tabela que associa todos os possíveis valores lógicos dos operandos em expressões com conectivos lógicos aos valores resultantes da expressão que usa este conectivo.

Para o conectivo lógico **não**, temos a seguinte tabela verdade:

a	nao a
verdadeiro	falso
falso	verdadeiro

Tabela verdade: não

A tabela verdade é uma tabela que associa todos os possíveis valores lógicos dos operandos em expressões com conectivos lógicos aos valores resultantes da expressão que usa este conectivo.

Para o conectivo lógico **não**, temos a seguinte tabela verdade:

a	nao a
verdadeiro	falso
falso	verdadeiro

Tabela verdade: e

A tabela verdade para o conectivo lógico e é:

a	b	a e b
verdadeiro	verdadeiro	
verdadeiro	falso	
falso	verdadeiro	
falso	falso	

Tabela verdade: e

A tabela verdade para o conectivo lógico e é:

a	b	a e b
verdadeiro	verdadeiro	verdadeiro
verdadeiro	falso	
falso	verdadeiro	
falso	falso	

Tabela verdade: e

A tabela verdade para o conectivo lógico e é:

a	b	a e b
verdadeiro	verdadeiro	verdadeiro
verdadeiro	falso	falso
falso	verdadeiro	falso
falso	falso	falso

Tabela verdade: e

A tabela verdade para o conectivo lógico e é:

a	b	a e b
verdadeiro	verdadeiro	verdadeiro
verdadeiro	falso	falso
falso	verdadeiro	falso
falso	falso	falso

Tabela verdade: e

A tabela verdade para o conectivo lógico e é:

a	b	a e b
verdadeiro	verdadeiro	verdadeiro
verdadeiro	falso	falso
falso	verdadeiro	falso
falso	falso	falso

Tabela verdade: ou

A tabela verdade para o conectivo lógico **ou** é:

a	b	a ou b
verdadeiro	verdadeiro	
verdadeiro	falso	
falso	verdadeiro	
falso	falso	

Tabela verdade: ou

A tabela verdade para o conectivo lógico **ou** é:

a	b	a ou b
verdadeiro	verdadeiro	verdadeiro
verdadeiro	falso	
falso	verdadeiro	
falso	falso	

Tabela verdade: ou

A tabela verdade para o conectivo lógico **ou** é:

a	b	a ou b
verdadeiro	verdadeiro	verdadeiro
verdadeiro	falso	verdadeiro
falso	verdadeiro	
falso	falso	

Tabela verdade: ou

A tabela verdade para o conectivo lógico **ou** é:

a	b	a ou b
verdadeiro	verdadeiro	verdadeiro
verdadeiro	falso	verdadeiro
falso	verdadeiro	verdadeiro
falso	falso	

Tabela verdade: ou

A tabela verdade para o conectivo lógico **ou** é:

a	b	a ou b
verdadeiro	verdadeiro	verdadeiro
verdadeiro	falso	verdadeiro
falso	verdadeiro	verdadeiro
falso	falso	falso

A ordem de prioridade para a resolução de expressões lógicas é a seguinte:

- 1 parênteses;
- 2 expressões numéricas;
- 3 operadores relacionais;
- 4 conectivos lógicos.

Um exemplo de expressão lógica, e sua resolução é:

$$4 + 3 * 5 \leq 3 \text{ e } 8 / 2 = 4 \text{ ou } \mathbf{\text{verdadeiro}} \text{ ou } 5 < 7$$

Um exemplo de expressão lógica, e sua resolução é:

$$4 + 3 * 5 \leq 3 \text{ e } 8 / 2 = 4 \text{ ou } \mathbf{\text{verdadeiro}} \text{ ou } 5 < 7$$
$$4 + 15 \leq 3 \text{ e } 8 / 2 = 4 \text{ ou } \mathbf{\text{verdadeiro}} \text{ ou } 5 < 7$$

Um exemplo de expressão lógica, e sua resolução é:

$$4 + 3 * 5 \leq 3 \text{ e } 8 / 2 = 4 \text{ ou } \mathbf{\text{verdadeiro}} \text{ ou } 5 < 7$$
$$4 + 15 \leq 3 \text{ e } 8 / 2 = 4 \text{ ou } \mathbf{\text{verdadeiro}} \text{ ou } 5 < 7$$
$$19 \leq 3 \text{ e } 8 / 2 = 4 \text{ ou } \mathbf{\text{verdadeiro}} \text{ ou } 5 < 7$$

Um exemplo de expressão lógica, e sua resolução é:

$$4 + 3 * 5 \leq 3 \text{ e } 8 / 2 = 4 \text{ ou } \mathbf{\text{verdadeiro}} \text{ ou } 5 < 7$$

$$4 + 15 \leq 3 \text{ e } 8 / 2 = 4 \text{ ou } \mathbf{\text{verdadeiro}} \text{ ou } 5 < 7$$

$$19 \leq 3 \text{ e } 8 / 2 = 4 \text{ ou } \mathbf{\text{verdadeiro}} \text{ ou } 5 < 7$$

$$19 \leq 3 \text{ e } 4 = 4 \text{ ou } \mathbf{\text{verdadeiro}} \text{ ou } 5 < 7$$

Um exemplo de expressão lógica, e sua resolução é:

$$4 + 3 * 5 \leq 3 \text{ e } 8 / 2 = 4 \text{ ou } \mathbf{verdadeiro} \text{ ou } 5 < 7$$

$$4 + 15 \leq 3 \text{ e } 8 / 2 = 4 \text{ ou } \mathbf{verdadeiro} \text{ ou } 5 < 7$$

$$19 \leq 3 \text{ e } 4 = 4 \text{ ou } \mathbf{verdadeiro} \text{ ou } 5 < 7$$

$$19 \leq 3 \text{ e } 4 = 4 \text{ ou } \mathbf{verdadeiro} \text{ ou } 5 < 7$$

$$\mathbf{falso} \text{ e } 4 = 4 \text{ ou } \mathbf{verdadeiro} \text{ ou } 5 < 7$$

Um exemplo de expressão lógica, e sua resolução é:

$$4 + 3 * 5 \leq 3 \text{ e } 8 / 2 = 4 \text{ ou } \mathbf{verdadeiro} \text{ ou } 5 < 7$$

$$4 + 15 \leq 3 \text{ e } 8 / 2 = 4 \text{ ou } \mathbf{verdadeiro} \text{ ou } 5 < 7$$

$$19 \leq 3 \text{ e } 8 / 2 = 4 \text{ ou } \mathbf{verdadeiro} \text{ ou } 5 < 7$$

$$19 \leq 3 \text{ e } 4 = 4 \text{ ou } \mathbf{verdadeiro} \text{ ou } 5 < 7$$

$$\mathbf{falso} \text{ e } 4 = 4 \text{ ou } \mathbf{verdadeiro} \text{ ou } 5 < 7$$

$$\mathbf{falso} \text{ e } \mathbf{verdadeiro} \text{ ou } \mathbf{verdadeiro} \text{ ou } 5 < 7$$

Um exemplo de expressão lógica, e sua resolução é:

$$4 + 3 * 5 \leq 3 \text{ e } 8 / 2 = 4 \text{ ou } \mathbf{verdadeiro} \text{ ou } 5 < 7$$

$$4 + 15 \leq 3 \text{ e } 8 / 2 = 4 \text{ ou } \mathbf{verdadeiro} \text{ ou } 5 < 7$$

$$19 \leq 3 \text{ e } 8 / 2 = 4 \text{ ou } \mathbf{verdadeiro} \text{ ou } 5 < 7$$

$$19 \leq 3 \text{ e } 4 = 4 \text{ ou } \mathbf{verdadeiro} \text{ ou } 5 < 7$$

$$\mathbf{falso} \text{ e } 4 = 4 \text{ ou } \mathbf{verdadeiro} \text{ ou } 5 < 7$$

$$\mathbf{falso} \text{ e } \mathbf{verdadeiro} \text{ ou } \mathbf{verdadeiro} \text{ ou } 5 < 7$$

$$\mathbf{falso} \text{ e } \mathbf{verdadeiro} \text{ ou } \mathbf{verdadeiro} \text{ ou } \mathbf{verdadeiro}$$

Um exemplo de expressão lógica, e sua resolução é:

$4 + 3 * 5 \leq 3$ e $8 / 2 = 4$ ou **verdadeiro** ou $5 < 7$
 $4 + 15 \leq 3$ e $8 / 2 = 4$ ou **verdadeiro** ou $5 < 7$
 $19 \leq 3$ e $8 / 2 = 4$ ou **verdadeiro** ou $5 < 7$
 $19 \leq 3$ e $4 = 4$ ou **verdadeiro** ou $5 < 7$
falso e $4 = 4$ ou **verdadeiro** ou $5 < 7$
falso e **verdadeiro** ou **verdadeiro** ou $5 < 7$
falso e **verdadeiro** ou **verdadeiro** ou **verdadeiro**
falso ou **verdadeiro** ou **verdadeiro**

Um exemplo de expressão lógica, e sua resolução é:

$4 + 3 * 5 \leq 3$ e $8 / 2 = 4$ ou **verdadeiro** ou $5 < 7$
 $4 + 15 \leq 3$ e $8 / 2 = 4$ ou **verdadeiro** ou $5 < 7$
 $19 \leq 3$ e $8 / 2 = 4$ ou **verdadeiro** ou $5 < 7$
 $19 \leq 3$ e $4 = 4$ ou **verdadeiro** ou $5 < 7$
falso e $4 = 4$ ou **verdadeiro** ou $5 < 7$
falso e **verdadeiro** ou **verdadeiro** ou $5 < 7$
falso e **verdadeiro** ou **verdadeiro** ou **verdadeiro**
falso ou **verdadeiro** ou **verdadeiro**
verdadeiro ou **verdadeiro**

Um exemplo de expressão lógica, e sua resolução é:

$4 + 3 * 5 \leq 3$ e $8 / 2 = 4$ ou **verdadeiro** ou $5 < 7$
 $4 + 15 \leq 3$ e $8 / 2 = 4$ ou **verdadeiro** ou $5 < 7$
 $19 \leq 3$ e $8 / 2 = 4$ ou **verdadeiro** ou $5 < 7$
 $19 \leq 3$ e $4 = 4$ ou **verdadeiro** ou $5 < 7$
falso e $4 = 4$ ou **verdadeiro** ou $5 < 7$
falso e **verdadeiro** ou **verdadeiro** ou $5 < 7$
falso e **verdadeiro** ou **verdadeiro** ou **verdadeiro**
falso ou **verdadeiro** ou **verdadeiro**
verdadeiro ou **verdadeiro**
verdadeiro